

FINAL REPORT
U.S. Department of Energy
'Advanced Sensing and Control Techniques to Facilitate
Semi-Autonomous Decommissioning of Hazardous Sites'

Lead Principal Investigator:

Robert J. Schalkoff
Dept. of Electrical and Computer Engr.
Clemson University,
Clemson, SC, 29634-0915
(864)-656-5913
rjschal@clemson.edu

Collaborators:

Robert. M Geist,
Dept. of Computer Science,
Clemson University
and
Darren Dawson,
Dept. of Electrical and Computer Engr.
Clemson University

Project Number: 55052
GRANT #DE-FG07-96ER14728:
Grant Project Officers:
Dr. Robert Price and Dr. Ramoncita Massey
Project Duration: 9/15/1996- 9/14/2000

1 Table of Contents

Contents

1	Table of Contents	1
2	Executive Summary	2
3	Research Objectives	2
4	Methods and Results	2
4.1	Vision Effort: Geometric and Photometric Scene Characterization Prior to Virtualization . . .	2
4.1.1	Overall Vision System Operation and Current Status	2
4.1.2	Vision Hardware	4
4.1.3	Geometric Characterization Algorithm Development and Software	4
4.2	Recalibration Under Sensor Package Motion	7
4.3	Integration w/ VR: Segmentation of Point Clouds into Individual Surfaces	7
4.4	Virtual Reality Environment Creation	8
4.4.1	Rendering Inversion.	8
4.4.2	Extracting Geometry	8
4.5	Robotic Thrust	14
4.5.1	Introduction	14
4.5.2	Overview of the Disassembly System	14
4.5.3	Software Components	14
5	Relevance, Impact and Technology Transfer	20
6	Personnel Supported	21
7	Publications	21
8	Interactions	23
9	Future Work	24

2 Executive Summary

This report summarizes work after 4 years of a 3-year project (no-cost extension of the above-referenced project for a period of 12 months granted).

- The fourth generation of a vision sensing head for geometric and photometric scene sensing has been built and tested. Estimation algorithms for automatic sensor calibration updating under robot motion have been developed and tested.
- We have modified the geometry extraction component of the rendering pipeline. Laser scanning now produces highly accurate points on segmented curves. These point-curves are input to a NURBS (non-uniform rational B-spline) skinning procedure to produce interpolating surface segments. The NURBS formulation includes quadrics as a sub-class, thus this formulation allows much greater flexibility without the attendant instability of generating an entire quadric surface. We have also implemented correction for diffuse lighting and specular effects.
- The QRobot joint level control was extended to a complete semi-autonomous robot control system for D&D operations.
- The imaging and VR subsystems have been integrated and tested.
- We have also had extensive discussions with INEEL personnel who have expressed interest in this work in conjunction with the DOE DDROPS effort. Two of the investigators (Schalkoff and Geist) visited INEEL in preparation for system integration.

Detailed technical descriptions of research progress to date are included in separate sections.

3 Research Objectives

This research is intended to advance the technology of semi-autonomous teleoperated robotics as applied to Decontamination and Decommissioning (D&D) tasks. Specifically, research leading to a prototype dual-manipulator mobile work cell is underway. This cell is supported and enhanced by computer vision, virtual reality (VR) and advanced robotics technology. The overall goal of this phase of the research is to autonomously generate a specification of scene geometry using a stereo pair of camera images sufficient to produce a virtual replica of the sampled scene that is suitable for VR rendering and subsequent real-time robotic D&D operations, navigation, planning, and mission training.

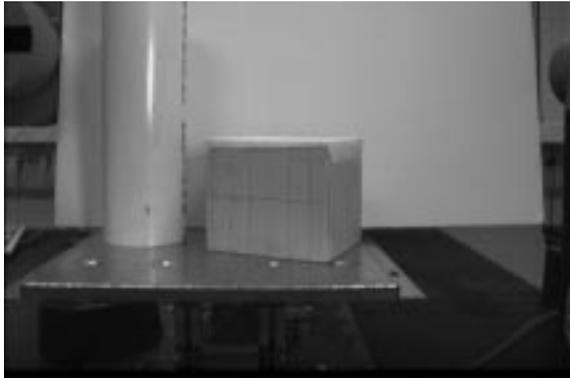
4 Methods and Results

4.1 Vision Effort: Geometric and Photometric Scene Characterization Prior to Virtualization

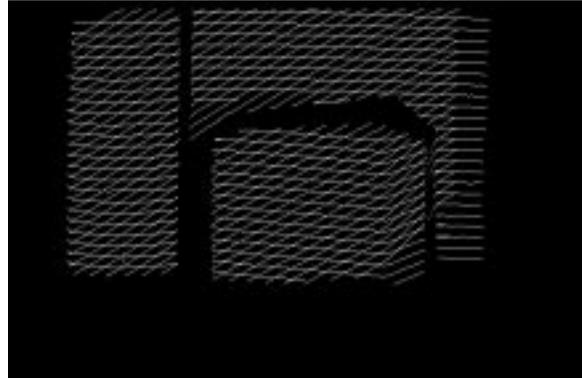
4.1.1 Overall Vision System Operation and Current Status

Basically, the vision system is used to acquire passive image data and generate surface (patch) models. This process is illustrated in Figure 1.

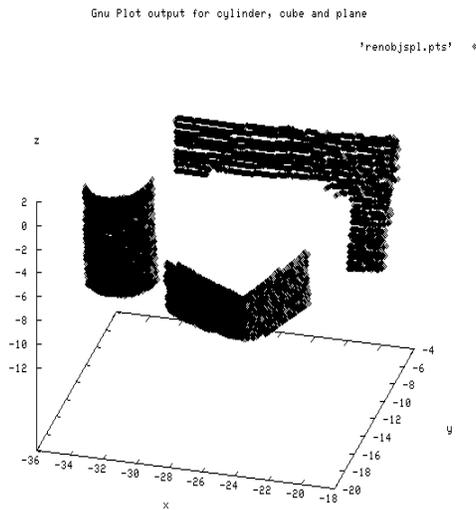
Four generations of a vision testbed for virtualization have been designed, fabricated and tested. The basic setup consists of a pair of monochrome cameras together with a laser stripe generator, and is described in more detail in the following sections. We have decided to deliver 3-D object points suitable for use with a nurbs-based surface representation instead of quadric surface models. Quadrics have limitations in the type



Gray-level rendition of initial scene (single viewpoint) consisting of 2 objects and (background) planar surface.



Composite of stripes used in geometric characterization.



3-D plot of estimated object points from different viewpoint.

Figure 1: Illustration of the Geometric Characterization of a Scene (prelude to virtualization).

of objects which may be represented as well as severe sensitivities to estimation in the presence of 3-D point noise.

Initial results indicate reasonable accuracy in mapping 3-D points with this testbed. A sample experiment is shown below. Further experiments are underway.

Current research efforts focus on the autonomous recalibration of the monochrome cameras following camera motion (with known end effector pose changes). The experimental element of this consists of mounting the vision sensing package on a PUMA arm and conducting additional experiments to establish calibration ability and to simulate recalibration following motion.

4.1.2 Vision Hardware

Our research effort involved four phases of a vision-based virtualization sensor. They were:

1. A laboratory pre-prototype w/ manual operation of laser.
2. A laboratory pre-prototype automatic laser scanning, and, to test system accuracy and enable recalibration, an associated movable "calibration plane".
3. A laboratory prototype mounted on a robotic arm and therefore configurable as a removable end effector tool.
4. A laboratory prototype which only requires manipulation of the laser via robot wrist motion. This is the design which is currently receiving the most attention.

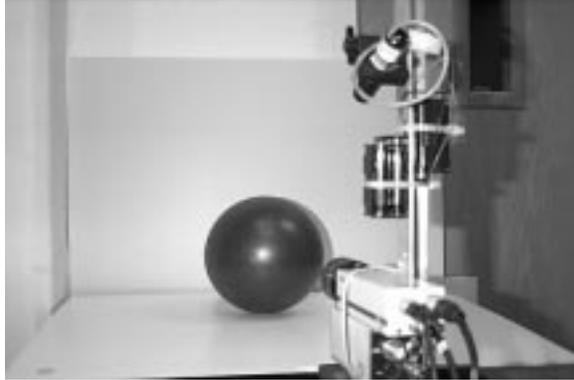
Figure 2 shows this evolution.

Active Camera. The active camera is a Lasiris 690nm, class IIIb laser projector with a straight-line generating lens. The laser line projects a sheet of light with a 30° fan angle from the lens [3]. In earlier designs, as shown in Figure 2, the laser is positioned by a Directed Perception PTU-46-70 high resolution pan-and-tilt unit. This high precision unit carries a step resolution of 46.2857 seconds arc per position. The pan-and-tilt unit provides motion with two degrees of freedom which allows the active camera to sweep the observed area with a cartesian or radial grid.

Monochrome Passive Cameras. The monochrome passive cameras are Pulnix TM-7CN devices fitted with a 12mm Cosmocar lens. The CCD imager has 768 horizontal by 494 vertical picture elements. The camera output is NTSC. This choice of camera has a peak response very near that of the human visual system. An interference filter is installed in the monochrome camera in order to record only visible light with a wavelength of 690nm \pm 2nm. This filter was chosen with respect to the frequency of the active laser source to increase laser stripe perception while minimizing noise due to ambient light.

4.1.3 Geometric Characterization Algorithm Development and Software

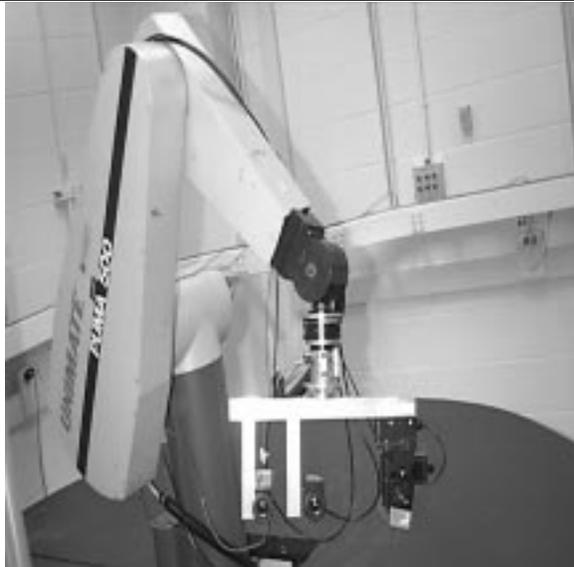
The 3-D model-to 2-D image transformation which maps the 3-D world into images (and inversions of this model) and the equations for multi-sensor vision including surface mapping are well known and well-studied [6]. In addition, there is no shortage of photogrammetric algorithms for deriving 3-D geometric information from sensed data. Often, some form of triangulation yields object points, which are then fitted to surface models. One of the oldest and still commonly used approaches for computer graphics is the fitting of polygon meshes derived from range data. Other examples include [8], [2], [7], [4] and [5]. Our virtualization approach employs a novel use of the bilinear constraint for geometric environment characterization



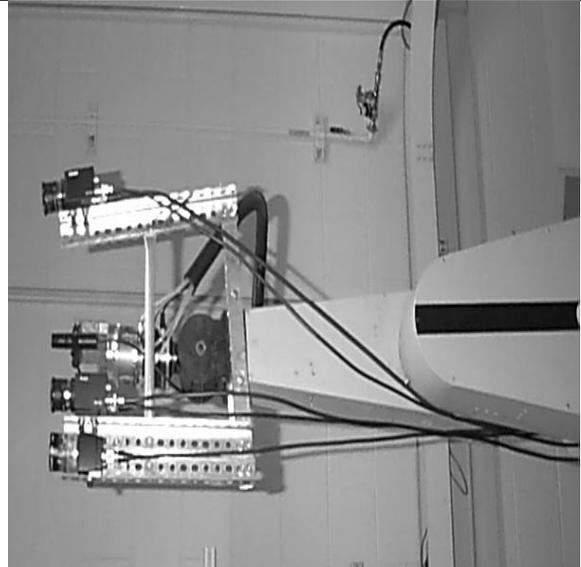
Early laboratory virtualization prototype used for "blue ball" experiment (mpeg available on web). Note crude laser manipulation capability.



First generation stationary virtualization head with movable laser and "moving planes" calibration unit together with simple 2-object scene and planar backgrounds (no longer in development).



Closeup of third-generation end-effector-mounted sensing package. In this format, the entire package is wrist-mounted. Laser motion is independent of wrist position using separate scanning motors.



Current virtualization sensing package. (In this format, the passive cameras are arm mounted and movable, but stationary with respect to the end effector since only the laser is wrist mounted. The system is no longer a replacable end-effector tool.)

Figure 2: Evolution of the Vision Head Used for Virtualization

Point Mapping Using "Active" Cameras Assume we fit a laser with optics such that a single line or "stripe" is projected. In space, this corresponds to a semi-infinite plane. Consider the process to get the 3D location $[x \ y \ z]^T$ of an arbitrary point on the illuminated stripe with image pixel at $[x_i \ y_i]^T$. When the passive camera is calibrated, the image point location provides 2 constraints

$$(a_{11} - a_{31}x_i)x + (a_{12} - a_{32}x_i)y + (a_{13} - a_{33}x_i)z = a_{34}x_i - a_{14} \quad (1)$$

$$(a_{21} - a_{31}y_i)x + (a_{22} - a_{32}y_i)y + (a_{23} - a_{33}y_i)z = a_{34}y_i - a_{24} \quad (2)$$

Another constraint comes from the (assumed known) projected plane where the illuminated point is constrained by

$$p_1x + p_2y + p_3z + p_4 = 0 \quad (3)$$

The projected plane parameter $[p_1 \ p_2 \ p_3 \ p_4]$ must be known precisely to make solving for $[x \ y \ z]^T$ successful.

Calibration of the active camera is difficult and time consuming. On the other hand, calibrating the passive sensors is relatively easy and fast. However, without a structured light source, it is difficult to solve the correspondence problem.

Using the Bilinear Constraint The stereo model described above can be viewed in several other ways. Using the above equations, the relationship in homogeneous equations is

$$[-P \ | \ \underline{F}]x_o = 0$$

Defining the 4×4 matrix C as

$$C = [-P \ | \ \underline{F}]$$

The above equation states that since $Cx_o = 0$. This is the basis for developing constraints on corresponding image point locations in a stereo vision system. The elements in C are functions of only a_{kij} (the calibration parameters for the two sensor system), and the corresponding image plane points $(x_{i1}, y_{i1}, x_{i2}, y_{i2})$. Also from the above equations, the determinant of C is constrained:

$$|C| = 0$$

Expanding this determinant and simplifying the result yields the important result that x_{i1}, y_{i1}, x_{i2} , and y_{i2} are constrained by the single equation

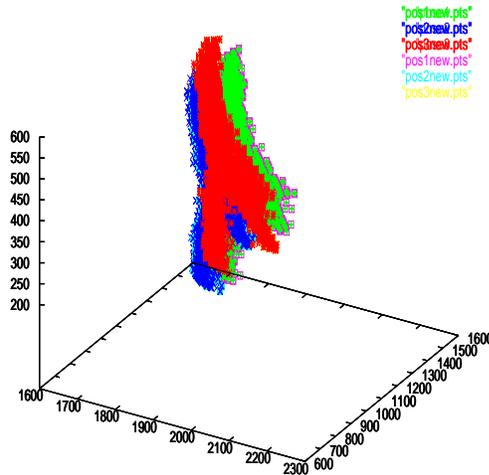
$$\begin{aligned} &x_{i1}x_{i2}m_1 + y_{i1}y_{i2}m_2 + x_{i1}y_{i2}m_3 + \\ &x_{i2}y_{i1}m_4 + x_{i1}m_5 + x_{i2}m_6 + y_{i1}m_7 \\ &+ y_{i2}m_8 + m_9 = 0 \end{aligned}$$

which is bilinear in nature. More importantly, **fixing an image point in one image plane constrains the locus of the other point to lie along a line**. This equation is referred to as the bilinear relationship or *bilinear constraint* in the variables x_i and y_i . The m_i coefficients in the above equation are, from the determinant expansion, functions of the a_{kij} sensor calibration parameters. The algebraic derivation, while straightforward, is quite tedious, especially if attempted by hand. Our approach is based upon a symbolic derivation in Maple.

4.2 Recalibration Under Sensor Package Motion

Sensor motion is necessary to enable a portable device which virtualizes large, complex buildings. Thus, in order to continue to generate accurate geometric information with the sensor, the calibration matrices must be autonomously updated. Thus, it is necessary to estimate the transformation between the robot's joint and the camera location. The general transformation matrix between the robot joint and the camera position (M) is given by the robot joints. The motion matrix will not be the same for the camera, hence the need to estimate the transformation matrix between the camera and the robot joints. This is the current emphasis of our work; sample results are illustrated below.

Object points estimated using recalibration techniques and moving head from 3 poses.



4.3 Integration w/ VR: Segmentation of Point Clouds into Individual Surfaces

As a prelude to generation of a 3-D VR rendering of the scene, individual surfaces must be identified. We have utilized 4 approaches to solving this problem:

1. A direct procedure based upon the lcm.
2. Estimation of quadric surface patch parameters for discrimination.
3. The use of "segmentation stripes", as shown in Figure 1.
4. NURBS representations (current effort).

References

- [1] M.J. Kraess. A vision module for the determination of surface parameters. Master's thesis, Clemson University, 1993.
- [2] S. Kumar, S. Han, D. Goldgof, and K. Bowyer. On recovering hyperquadrics from range data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:1079–1083, November 1995.
- [3] Lasiris Corporation. *Diode Laser Products Instruction Manual*.
- [4] N.S. Raja and A. K. Jain. Obtaining generic parts from range data using a multi-view representation. *GVGIP:Image Understanding*, 60:44–64, July 1994.

- [5] K. Rao and R. Nevatia. Computing volume descriptions from sparse 3-d data. *Int. J. Computer Vision*, 2:33–50, 1988.
- [6] R. J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley & Sons, New York, 1989.
- [7] F. Solina and R. Bajscy. Recovery of parametric models form range images: The case for superquadrics with global deformation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(2):170–176, 1990.
- [8] K. Wu and M.D. Levine. 3-d shape approximation using paramteric geons. *Image and Vision Computing*, 15:143–158, 1997.

4.4 Virtual Reality Environment Creation

4.4.1 Rendering Inversion.

The goal of this task is to receive input that consists of a specification of scene geometry, a stereo pair of camera images, and the associated camera calibration matrices and to produce from these a virtual replica of the sampled scene that is suitable for real-time navigation, planning, and mission training.

Rendering Virtual Environments. The constraint of real-time display precludes sophisticated treatment of specular information, but it allows global illumination effects through the ambient and diffuse components. A suitable rendering equation, in terms of radiance, is thus of the form

$$c = B/\pi + k_s(H \bullet N)^m \quad (4)$$

where B denotes radiosity (exiting irradiance), computed through a classical formulation of environment patch interaction, and the remainder is a layered first-order specular component whose magnitude is determined by the surface normal, N , and the vector, H , that points half-way between light source and eye position [2]. The goal is to use automatically gathered radiometric and geometric information from a real environment to recover the necessary parameters for using (4) to produce a virtual replica.

Radiosity offers an adequate treatment of ambient light and diffuse (Lambertian) reflections, and, key to real-time interaction, it provides a view-independent lighting solution. We use a standard formulation [1]. Every object in an environment is specified in terms of discrete patches. The radiosity (exiting irradiance) of patch i , B_i , is:

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j \frac{A_j}{A_i} F_{ji} \quad (5)$$

where E_i = emission of patch i , F_{ji} = form factor from j to i , A_i = area of patch i , ρ_i = reflectivity (bi-hemispherical reflectance) of patch i , and n = number of discrete patches.

For surfaces that are to be texture mapped, the radiosity computation must reflect the texture's approximate reflectivity to other patches. Also, at the end of the radiosity preprocessing phase, the radiated patch colors must be modified so that the application of the texture blend function will yield the correct result. This amounts to replacing patch exiting irradiance with incident irradiance, $(B_i - E_i)/\rho_i$ (see [4]).

4.4.2 Extracting Geometry

Key to rendering inversion is the availability of scene geometric information. The computational foundations of our object recognition technique are based on structured light projection across geometries of known classes. In the initial phases of this effort, we assumed that surface patches were quadrics, i.e.,

$$\{\vec{x} | \vec{x} Q \vec{x}^T = 0\} \quad (6)$$

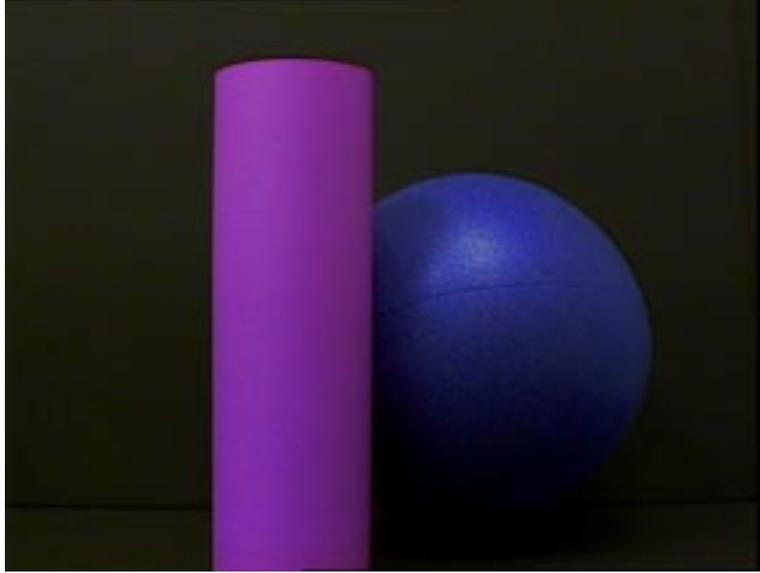


Figure 3: Photograph of test scene.

where $\vec{x} = (x, y, z, 1)$ and Q is a real, symmetric matrix of the form:

$$Q = \begin{pmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{pmatrix} \quad (7)$$

This assumption remains a reasonable one for the target industrial environments, where more than 90% of machined parts are bounded by quadric surfaces [3].

It can be shown that projection of a laser stripe across the patch necessarily yields a quadratic curve,

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (8)$$

in a (perspective) passive camera image plane. The six coefficients of the quadratic curve are insufficient to determine the ten coefficients of the quadric, but a second, independent laser stripe completely determines the surface [5].

Nevertheless, it turns out that quadric surface shape is a relatively unstable function of the estimates of the entries in Q . Consider the test scene (photograph) shown in figure 3. A virtual replica, automatically constructed using the techniques described herein, is shown in figure 4. Although this replica is satisfactory from the view shown (a replicated camera view) and this replica is a 3-D virtual environment that may be traversed in real-time, we find the recognized geometry at considerable variance with the real geometry. In figure 5 we show a wireframe version of the virtual replica from another view. Although the real scene objects were a cylinder and a sphere, both were recognized as ellipsoids, which was only locally correct.

As a result, we have modified the geometry extraction component of the rendering pipeline. Laser scanning now produces highly accurate points on segmented curves. These point-curves are input to a NURBS (non-uniform rational B-spline) skinning routine [6] to produce interpolating surface segments. The segments do not extend beyond the range of the point collection, but, since the NURBS formulation includes quadrics as a sub-class, this formulation allows much greater flexibility without the attendant instability of generating an entire quadric surface.

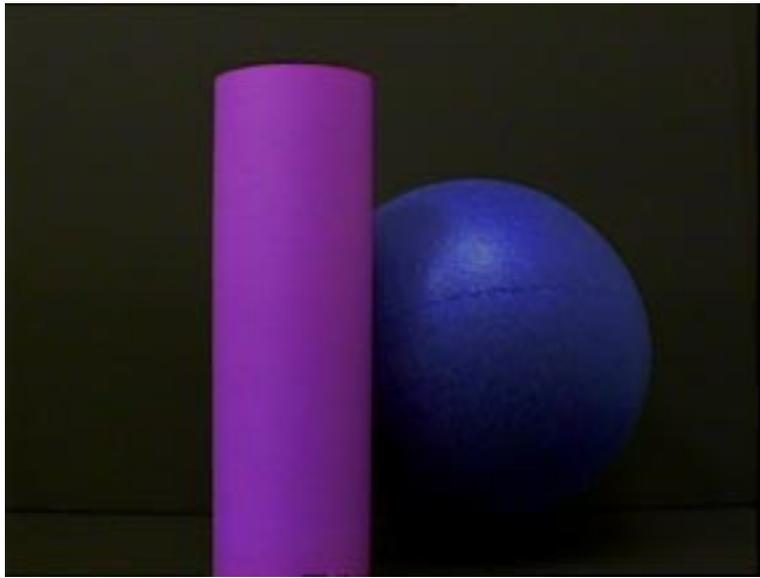


Figure 4: Virtual replica of test scene.

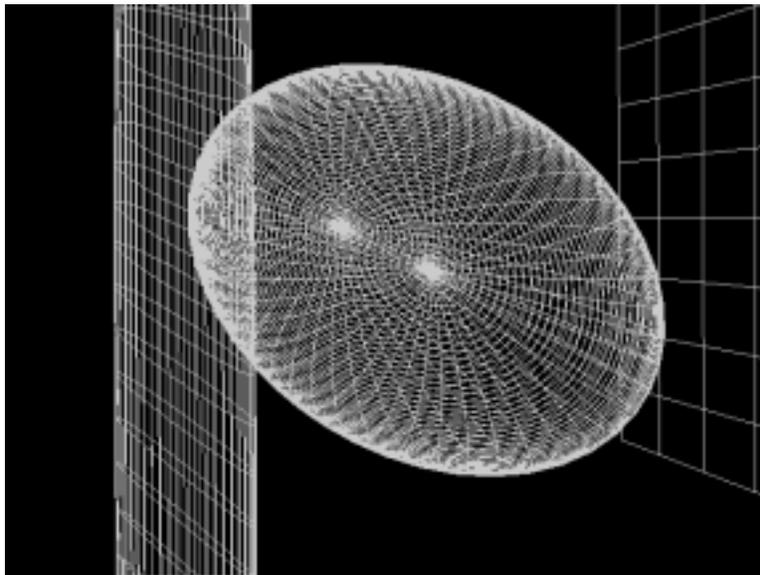


Figure 5: Virtual replica from another view.



Figure 6: Target test image.

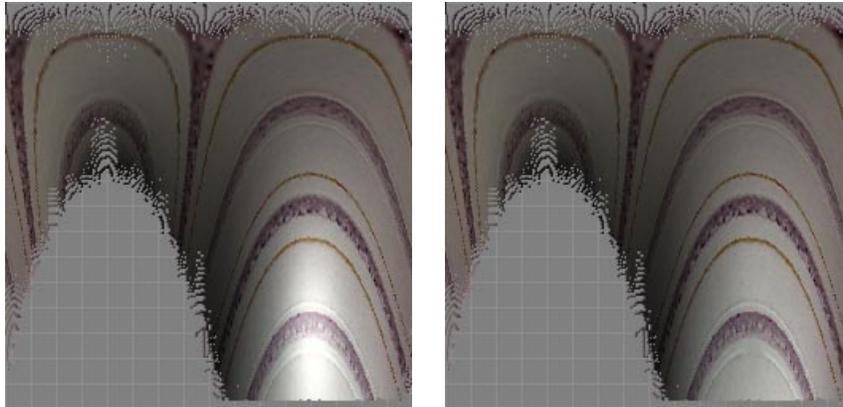


Figure 7: Texture extraction.

Extracting Color and Texture A single stereo photograph, together with scene geometry estimates from the structured light projection, can be used to extract the parameter estimates required for (4). In particular, we first estimate k_s and m and then extract reflectivity, ρ . The remaining terms of (4) can then be deduced from the geometry.

We begin by extracting a raw texture sample from each object in the scene. Using the estimated geometry, we build a virtual replica in wireframe, and attach “rainbow” textures to the virtual objects. These “rainbow” textures have a unique color per pixel for sampling identification. Given a real scene image, with known camera position and light source, we generate a virtual image of the corresponding “rainbow-textured” objects from the same viewpoint. This “rainbow” image serves as a collection of masks that allow texture sampling from the real scene image. Of course, unless the object is planar, the texture sample will not be completely filled, and, in any case, the sample will be distorted by lighting effects in the real scene.

In figure 6 we show a (synthetic) photograph as an example target image. This is the left-eye view of a stereo pair. A raw texture sample, taken from the target striped ellipsoid is shown in the left image of figure 7.

Specular Correction

Those pixels visible from both the eye point and the light source in a stereo pair of images provide us with sample data from which we can deduce specular information. For each filled pixel in a (left) texture sample, we can apply the inverted camera calibration matrix to find a world-coordinate ray that projects to that pixel. Using the known geometry, we can find the world coordinates of the ray/surface intersection and then apply the (right) camera calibration matrix to find the corresponding pixel in the (right) photo of the stereo pair. Each pixel pair provides us with a sample whose difference in (4) eliminates the radiosity component, i.e.,

$$c_l - c_r = k_s((H_l \bullet N)^m - (H_r \bullet N)^m) \quad (9)$$

From the collection of such samples, we can fit k_s and m to minimize squared deviation. Given these parameters, we can then remove specular effects from the sample. In right image of figure 7, we show the same texture sample with specular effects removed. During rendering, specular highlights are re-applied by using the removed information as a dynamically applied texture map [2].

Diffuse Correction

Correction for diffuse lighting is much more difficult. Although direct illumination of vertices might be expected to vary simply with $L \bullet N$, this will be inadequate for scenes with significant indirect illumination, as commonly found in real scene images.

Ideally, we seek an inversion of the radiosity equation (5), i.e.,

$$\rho_i = \frac{B_i - E_i}{\sum_{j=1}^n B_j \frac{A_j}{A_i} F_{ji}} \quad (10)$$

An immediate problem with this approach is that a complete solution of (10) begs the question: if, from samples, we have accurate estimates for all components of the radiosity vector, B , we have no need for calculation of the bi-hemispherical reflectance, ρ . We can simply use the known radiosity values directly! The catch, of course, is that sampling the environment at this level is beyond reach. The summands in the denominator of (10) are only known (estimated) for those patches j that appear in the passive image(s). Thus we have incomplete information, and we must estimate the contribution of the missing components.

To do this, we observe that the target radiosity and unknown reflectance for any surface are always related by $B \leq \rho \leq 1.0$. Thus we can iterate radiosity steps in performing a binary search for ρ over the bounding interval. The scene photos serve as the control. Tests indicate that fewer than 10 radiosity steps are needed for convergence [7].

Extrapolation

During the diffuse recovery process, partially sampled textures must be extrapolated to full surface textures in order to estimate the diffuse reflectance of scene elements not visible in the stereo photo pair.

If we assume texture patterns continue across non-visible parts of the object, then a discrete cosine transform (DCT) offers a reasonable solution. In particular, we find a large pixel rectangle that is completely sampled in the partial texture map, and compute the DCT thereon. If the pixel rectangle values, $p_{x,y}$, are in the range 0.0 - 255.0 per channel for $x \in \{0, 1, \dots, M-1\}$ and $y \in \{0, 1, \dots, N-1\}$, then the transform is:

$$P_{u,v} = \frac{4C_u C_v}{MN} \sum_{x=0}^M \sum_{y=0}^N (p_{x,y} - 128) \cos(((2x+1)u\pi)/(2M)) \cos(((2y+1)v\pi)/(2N)) \quad (11)$$

where $C_x = 1/\sqrt{2}$, if $x = 0$, and $C_x = 1$ otherwise. The inverse transform,

$$p_{x,y} = 128 + \sum_{u=0}^M \sum_{v=0}^N P_{u,v} C_u C_v \cos(((2x+1)u\pi)/(2M)) \cos(((2y+1)v\pi)/(2N)) \quad (12)$$

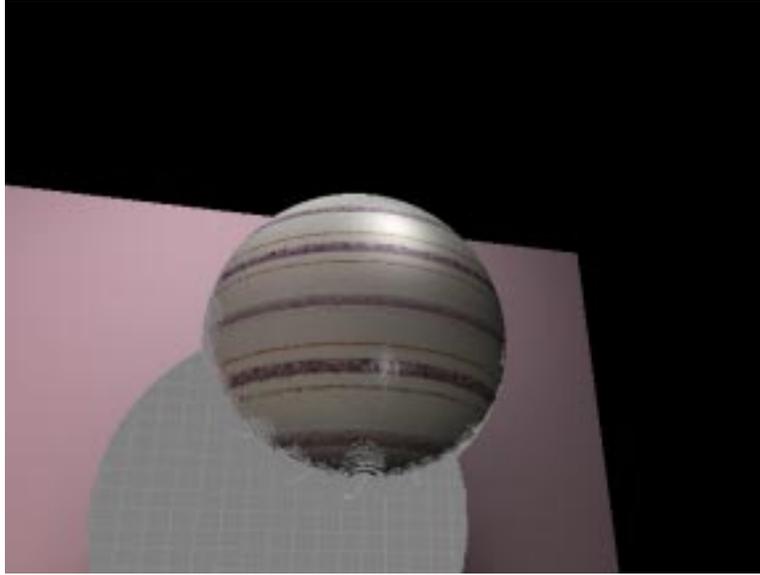


Figure 8: Virtual reconstruction (different view).

will reconstruct $p_{x,y}$ exactly in the initial rectangle, but we can assume that the inverse transform applies outside this range as well in order to generate pixel values for a complete texture map. This approach avoids introducing artificial boundaries or blurring.

In figure 8 we show a virtual replica of the scene of figure 6 reconstructed from samples as described. The view point has been changed, and the ellipsoid has been rendered with non-extrapolated textures in order to show the part of the ellipsoid that was initially invisible and hence is conjectured. Note that the specular highlight has moved, as is appropriate.

References

- [1] M. F. Cohen and J. R. Wallace. *Radiosity and realistic image synthesis*. Academic Press Professional, Cambridge, MA, 1993.
- [2] R. Danforth, A. Duchowski, R. Geist, and E. McAliley. A platform for gaze-contingent virtual environments. In *Proc. of the AAAI Smart Graphics Symp.*, Stanford, California, March 2000 (to appear).
- [3] F. T. Farago. *Handbook of Dimensional Measurement*. Industrial Press, New York, NY, second edition, 1982.
- [4] R. Geist and O. Heim. Towards photorealism in virtual environments. In *Proc. of the 34th Annual ACM Southeast Conf.*, pages 125 – 132, Tuskegee, Alabama, April 1996.
- [5] R. Geist, R. Schalkoff, T. Stinson, and S. Gurbuz. Autonomous virtualization of real environments for telepresence applications. *PRESENCE: Teleoperators and Virtual Environments*, 6(6):645–657, 1997.
- [6] L. Piegl. On nurbs: A survey. *IEEE CG&A*, 11:55–71, 1991.
- [7] D. Vernon. Rendering virtual environments from photographs: An autonomous, image-based approach for telepresence applications. Master’s thesis, Dept. of Comp. Sci., Clemson Univ., 1998.

4.5 Robotic Thrust

4.5.1 Introduction

The QRobot joint level control [2] was extended to a complete semi-autonomous robot control system for D&D operations. The QRobot system is a purely PC based system that integrates the following components: *i)* joint level control and a trajectory generator with a high level programming interface for Puma manipulators, *ii)* a 3D OpenGL-based hardware-accelerated robot simulator, *iii)* video based and VR based operator interfaces, *iv)* teleobservation programs, *v)* the ability to interface with a variety of sensors, and *vi)* control of different robotic end-effectors. The experimental section of this section of the report documents the capability of the QRobot system to perform a sample D&D operation.

4.5.2 Overview of the Disassembly System

Figures 9, 10, and 11 show the software and hardware components that are distributed across three PCs. The VR Operator Interface PC runs Windows NT, while the Robot Control PC and the WebCam PC run QNX, a real-time OS. The VR Operator Interface and the Robot Simulator are integrated into one Windows NT program. The Video Operator Interface contains the actual disassembly program and communicates with the VR Operator Interface over Internet Domain TCP/IP sockets.

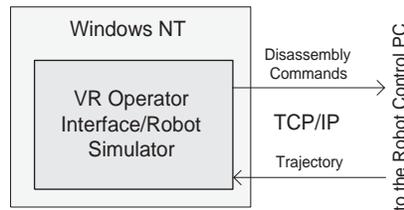


Figure 9: The VR Operator Interface PC

The Disassembly Program issues high level commands to ARCL, a robot control library that serves as the programming interface and as the trajectory generator. ARCL generates a stream of setpoints that are fed into the Joint Level Control. Observation Windows provide visual feedback of the D&D operation. They show a continuously updated image from one of the video cameras. Multiple observation windows can be used with different video cameras. The WebCam System allows video feedback over the World Wide Web by using a standard web browser. The camera, the pan/tilt unit (PTU) and the zoom lens of the WebCam System are also accessible from the observation windows running on the Robot Control PC. Special programs, called Hardware Servers, are responsible for accessing the PC boards (*e.g.*, PTU server, MultiQ server, etc.) Control Servers implement control algorithms (*e.g.*, zoom lens control, robot joint level control). Applications use Clients to communicate with servers.

4.5.3 Software Components

Multitasking and Communication Architecture: The system's functionality is split into many cooperating tasks. For these tasks to work seamlessly together, the OS must fulfill certain requirements. It must provide priority based deterministic CPU scheduling to ensure that high priority real-time tasks (*e.g.*, the joint level control) are not delayed by low priority non real-time tasks (*e.g.*, GUIs). It also must provide robust interprocess communication (IPC) mechanisms so that the cooperating tasks can synchronize and communicate. The real-time microkernel based OS QNX, developed by QSSL [11], meets all of these requirements. Unlike real-time extensions such as RT-Linux or Hyperkernel for Windows NT, QNX is a true microkernel real-time OS. One benefit of this is that the whole spectrum of OS functions, including file access and networking, can be used in real-time tasks.

Joint Level Control: The joint level control is implemented as a QNX program. This is a very flexible solution, since the control algorithm can be modified directly by changing and recompiling the control

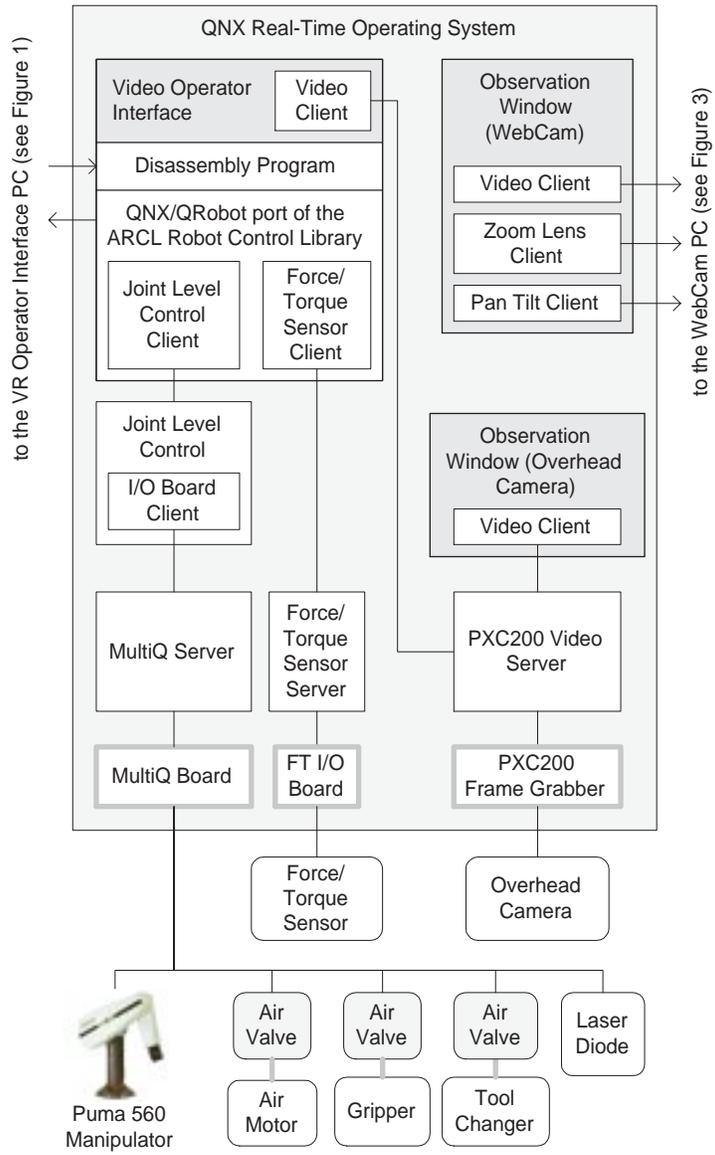


Figure 10: The Robot Control PC

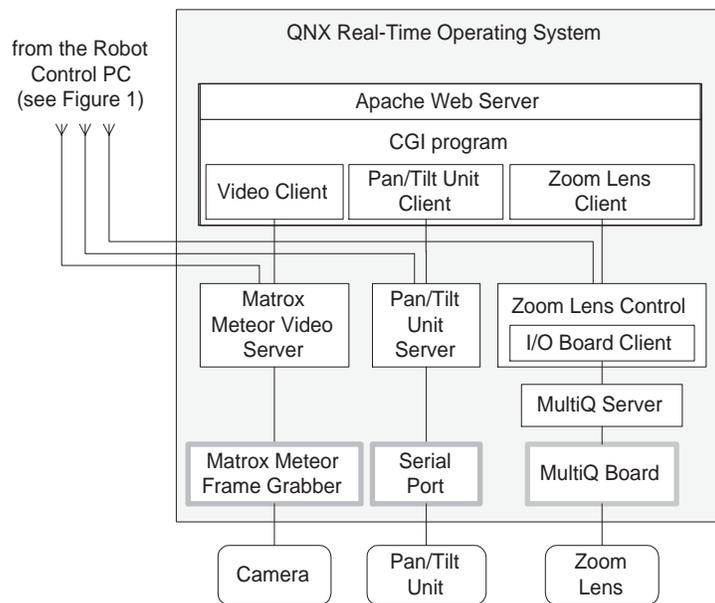


Figure 11: The WebCam PC

program. The ever-increasing computing power of PCs allows the implementation of more complex control algorithms. In addition, it is now possible to include arbitrary sensor information in the joint level control loop, which allows the implementation of advanced sensor-based control algorithms such as force-based control or direct visual servoing. The joint level control used in the D&D system was developed using QMotor [6], which is an environment for PC based control program development and implementation. The control program implements a PD controller with static and coulomb friction compensation for all joints and gravity compensation for the second and third joint. Joint velocities are calculated via a backward difference method. Since the backward difference method introduces noise, the resulting velocity signal is passed through a low pass filter [2]. The joint level control program works as a server and receives the stream of setpoints via message passing from the joint level control client, which is part of the QRobot version of ARCL. The control can be switched to a zero gravity mode. In this mode, the control only compensates for the gravity on the robot links instead of servoing to desired setpoints [12]. The robot can be freely moved by hand in this mode, which is used to teach end-effector positions and orientations with the teachpendant program.

Robotic Utility Programs: To facilitate the use and calibration of the D&D system, a set of robot utility programs was developed. These utility programs are described below.

- **Teachpendant:** To learn the end-effector positions and orientations used in the D&D operation, a teachpendant program was developed, see Figure 13. The teachpendant uses the zero-gravity mode of the joint level controller, which allows the user to easily push the robot around in the workspace. Once a position and orientation is found, it can be stored under a given name in a position list. It is possible to leave teach mode at any time and move the robot to previously taught positions. The position list can be stored in an ASCII file for later use in the teachpendant, or for use from an ARCL program.
- **PotVal:** The PotVal utility performs the initial calibration procedure of the Puma 560 that relates joint potentiometer readings to encoder readings.
- **PumaCal:** The PumaCal utility performs encoder calibration after power up of the manipulator. It determines the current position of the robot by using potentiometers and index pulses. This utility is similar to RCCL's pumacal utility [4], but performs the calibration in only a fraction of the time.

- **Disassembly Program:** A motor is used to demonstrate a simple disassembly. The objective is to remove the cap of the motor. Figure 14 shows the steps performed by the system: 1). Remove the first bolt. The gator grip is used to unscrew the bolt. Since the bolt usually stays in the housing, the operator has the additional option to remove the unscrewed bolt with the gripper and drop it into a container. 2). Remove the second bolt in the same fashion as described in step 1. 3). Perform a torch cut. In the experiment, the torch is simulated by a laser diode. 4). Remove the cap with the gripper and put it on the table. The disassembly program is written in C++. For each disassembly step, via points are determined with the teachpendant program and saved to a file. The disassembly program reads this file and creates transformations and position equations for each via point. The position equations are the input to the ARCL move function calls. Each disassembly step consists of picking up the right tool from the tool rack, performing the operation, and returning the tool to the rack. Some special functions are defined to allow the operator to intervene in case the system fails to complete a step. These functions include manually getting or returning a tool and manually locking or releasing a tool.
- **Operator Programs:** Four operator interface programs offer different control and feedback functions. Observation windows and the video operator interface run on the same PC as the robot control, but at a lower priority. They use Photon, the windowing system for QNX. Photon provides functionality similar to the X Window System and Xt. To accelerate GUI development under Photon, a C++ class library (CPhoton) was developed.
- **Observation Windows:** The observation window (see Figure 15) provides live visual feedback from a video camera. When using the WebCam, the observation window offers additional functionality. Clicking in the image centers the PTU about that point. The buttons at the bottom of the window control the WebCam's zoom lens. It is possible to start multiple observation windows and connect them to the same or different cameras. Since the observation windows use message passing based client/server communication, the camera servers can be distributed over multiple PCs. The disadvantage of message passing is reduced speed in the image transfer. Depending on the PC's performance, image size, image color depth, network traffic, and the video display driver, the observation window displays 1-5 frames per second.
- **Web Camera:** The WebCam is a World Wide Web based visual feedback, with similar functionality to the observation windows. The Apache web server starts a CGI program whenever the web page is accessed. The request for the web page contains the desired pan/tilt angles and the desired zoom factor as parameters. The CGI program moves the PTU to the desired position, sets the zoom factor, and captures an image. This image is converted to JPEG format, and a web page is dynamically created to show the image. The client/server architecture allows multiple observation windows and any number of web browsers to request images at the same time. The advantage of the WebCam is the accessibility from any Internet connected computer. The disadvantage is the lack of continuous and fast updates of the image.
- **Video Operator Interface:** The video operator interface provides even unskilled operators an easy-to-use interface to control the disassembly tasks. The video image is used to trigger disassembly operations. The operator moves the mouse cursor over a certain part of the motor that he wants to disassemble. The operator interface then displays a pop-up menu with a list of disassembly options. For example, when the operator moves the mouse over the motor end cap, the end cap is highlighted, and a menu pops up with the menu item "Remove Cap" . After the operator selects an operation, the program begins to perform the task and shows progress information in a dialog. As the disassembly is being performed, the operator is able to supervise the operation in the observation windows. In case the disassembly of a part is unsuccessful, the operation can be repeated. The image-based selection of disassembly operations is convenient for the operator, but it also requires that the system know where the parts of the object are located in the image. The Image Processing group investigated the use of advanced image processing and 3D-object virtualization techniques to automatically identify and locate these parts for the disassembly task [3].

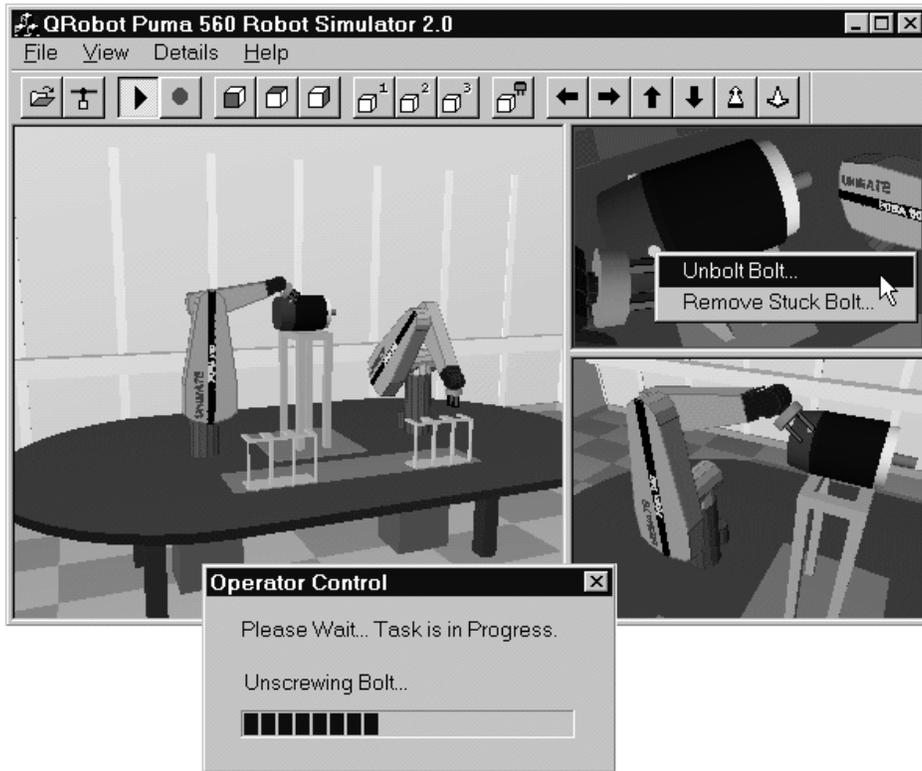


Figure 12: The Virtual Operator Interface/Robot Simulator



Figure 13: The QRobot Teachpendant

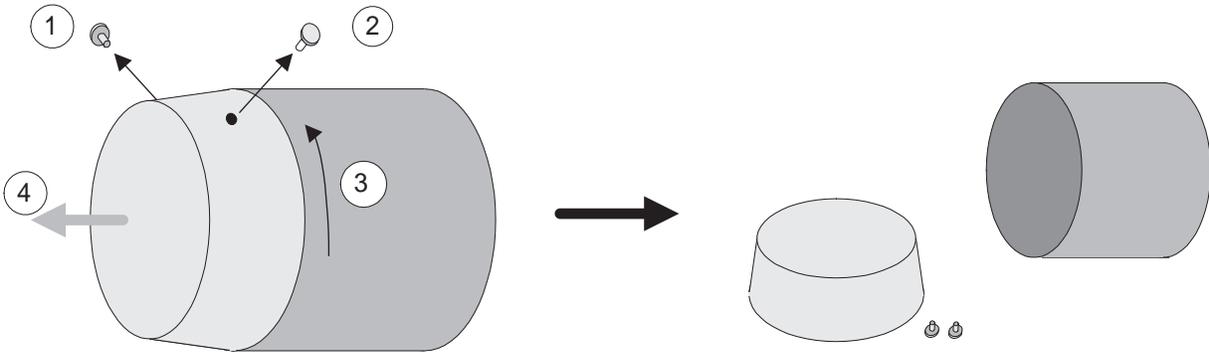


Figure 14: Steps to Disassemble the Motor



Figure 15: Observation Window

References

- [1] Vladimir Lumelsky, "On human performance in telerobotics", *IEEE Transactions on Systems, Man and Cybernetics*, 21(5).
- [2] N. Costescu, M. Loffler, E. Zergeroglu, and D. Dawson, "QRobot - A Multitasking PC Based Robot Control System", *Microcomputer Applications Journal Special Issue on Robotics*, Vol. 18 No. 1, pages 13-22.
- [3] <http://ece.clemson.edu/iaal/doeweb/doeweb.htm>.
- [4] J. Lloyd, *Implementation of a Robot Control Development Environment*, Masters Thesis, McGill University, Dec.1985.
- [5] Unimation Inc., Danbury, Connecticut, "500 Series Equipment and Programming Manual", 1983.
- [6] N. Costescu, D. M. Dawson, and M. Loffler, "QMotor 2.0 - A PC Based Real-Time Multitasking Graphical Control Environment", *IEEE Control Systems Magazine*, Vol. 19 Number 3 pages 68 - 76.
- [7] D. Hildebrand, "An Architectural Overview of QNX", *Proceedings of the Usenix Workshop on Micro-Kernels & Other Kernel Architectures*, Seattle, April 1992.
- [8] P. Corke, *The Unimation Puma Servo System*, CSIRO Division of Manufacturing Technology, Preston, Australia.
- [9] Trident Robotics and Research, Inc., 2516 Matterhorn Drive, Wexford, PA 15090-7962, (412) 934-8348, <http://www.cs.cmu.edu/~deadslug/puma.html>.

- [10] Quanser Consulting, 102 George Street, Hamilton, Ontario, CANADA L8P 1E2, Tel: 1 905 527 5208, Fax: 1 905 570 1906, <http://www.quanser.com>.
- [11] QSSL, Corporate Headquarters, 175 Terence Matthews Crescent, Kanata, Ontario K2M 1W8 Canada, Tel: +1 800-676-0566 or +1 613-591-0931, Fax: +1 613-591-3579, Email: info@qnx.com, <http://www.qnx.com> (QNX web site).
- [12] B. Armstrong, O. Khatib, J. Burdick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", *Proc. IEEE int. conf. Robotics and Automation*, 1986, pp. 510-518.
- [13] P. Corke, *The ARCL Robot Programming System*, CSIRO Division of Manufacturing Technology.
- [14] P. Corke, P. Dunn, and R. Kirkham, *An ARCL Tutorial*, CSIRO Division of Manufacturing Technology, Preston, Australia, 1992.
- [15] J. Neider, T. Davis, and M. Woo, *OpenGL Programming Guide*, Addison-Wesley Publishing Company, New York, 1993.

5 Relevance, Impact and Technology Transfer

The DOE is looking for new and innovative technologies that allow D&D operations to be faster, safer, and more cost-effective. Furthermore, the DOE complex has over 60,000 structures that require decontamination and decommissioning over the next 20 to 30 years. Some of the facilities contain radiation or toxic material hazards that make human entry unsafe, undesirable, or uninhabitable for extended periods (hours). Consequently, decontamination projects for these facilities could benefit by incorporating telerobotic robotic systems

Telerobotic systems provide a good solution to this problem. They allow robots to be remotely controlled from an operator console and provide visual feedback to the operator. In basic systems, an operator controls the robot directly (*e.g.*, with a joystick) and receives video feedback [1]. Performing a remote disassembly is a complicated, often repetitive task, which requires skilled operators. Therefore, much of the ongoing research focuses increasingly on the development of semi-autonomous systems. These systems perform higher level tasks, such as removing a bolt, triggered by the operator. Furthermore, VR based operator interfaces are desired to simplify interaction with the system.

The integration of imaging, robotics and virtual reality subsystems into highly efficient telerobotic robotic systems has been successful. While not complete, the research has demonstrated the feasibility and the actual progress on imaging systems is excellent.

The technical capability shown has a strong relevance to DOE's D&D mission, specifically the DDROPS and HANS55 programs (see Section ??). Tasks such as pipe cutting, concrete cutting, and dismantlement of structures are envisioned. Furthermore, the virtualization capability provided may be integrated with 3-D radiometric (radiation) mapping packages and is adaptable to different robotic end-effectors.

This research fits within the D&D investment portfolio as an Engineering Science effort within the Basic Energy Sciences (BES) program. The technical capability developed to date has a strong relevance to DOE's D&D mission, specifically the DDROPS and HANS55 programs. To this end, the Principal Investigator has had extensive conversations with engineers at INEEL; a letter indicating site interest in our work is contained on the next page.

Direct Relevance to Primary DOE Needs. There are at least five specific primary needs identified by DOE that this research directly addresses. These needs, together with DOE-supplied "Needs Identifiers" are:

1. Decontamination of Large Metallic Vessels Contaminated with TRU (AL-00-01-15-MW)
2. Radiation Hardened Robotics for Building 324 (RL-DD010)
3. Remote/Robotic Technologies for Access and Deployment of Characterization and Sampling Tools. (ID-7.2.19)

4. Robotics for D & D (ID-7.2.08)
5. Tritium Robotics (OH-M010)

6 Personnel Supported

- Faculty:

- Robert Schalkoff, Professor, Dept. Electrical and Computer Engr.
- Robert Geist, Professor, Dept. of Computer Science (supported)
- James Westall, Professor, Dept. of Computer Science (associated)
- Darren Dawson, Professor, Dept. Electrical and Computer Engr.

- Graduate students:

Ph.D. Students

- N. Costescu, *Real Time Control Environments for Mechatronic Systems*, graduation date: December 2000.
- E. Zergeroglu, *Model Based Control Algorithms for Robot Manipulators*, graduation date: December 2000.
- M. Loffler, *Real Time Control Platforms for Robot Manipulators*, graduation date: August 2001.
- Chakrit Watcharopas (supported); current Ph.D. student in Computer Science

Masters Thesis Students

- M. Steele, *Real Time Software for Control and Telepresence*, graduation date: May 2000.
- Z. Yao, *Real-Time Linux Target: A MATLAB-Based Graphical Control Environment*, graduation date: August 2000.

- Robert Danforth (associated); received M.S. in Computer Science; now at Micrografx
- David Vernon (supported); received M.S. in Computer Science; now at WareOnEarth
- S. Gurbuz, (M.S., EE), "Adaptive Image Segmentation Using Pipelined Hardware", December 1996.
- S. Mahoney, (M.S., EE), "Vision Head Design for Quadric Parameter Estimation", December 1998.
- C. Rumpf, (M.S., CpE), "Implementation of Quadric Parameter Estimation Algorithms", December 1998.
- Yu Saiyue (M.S., EE), "Autonomous Virtualization of Quadric Environments", August 1999.
- Raghavan Venugopal, (M.S., CpE), "Recalibration of Moving Sensors for Autonomous Virtualization", May 2001.

7 Publications

- am Ende, B., Geist, R., Schalkoff, R., Soukup, J., Westall, J., and Wright, D., "Surface Reconstruction with Data from Hazardous Environments," in preparation.
- Watcharopas, C., and Danforth, R., "Geometry and Reflectance Estimation for Virtualization," *Proc. 38th Annual ACM SE Conf.*, Clemson, SC, April, 2000, pp. 220 - 229.
- Danforth, R., and Geist, R., "Automatic Mesh Refinement and Its Application to Radiosity Computations," *Int. Journal of Robotics and Automation* **15**:1 (2000), Acta Press, pp. 1 - 8.

- Danforth, R., Duchowski, A., Geist, R., and McAliley, E., "A Platform for Gaze-Contingent Virtual Environments," *Proc. of the AAAI Smart Graphics Symp.*, Stanford, California, March, 2000, pp. 66 - 70.
- Danforth, R. and Geist, R., "A Spring-loaded Vertex Model for Automatic Surface Meshing," *Proc. IASTED Int. Conf. on Computer Graphics and Imaging (CGIM'99)*, Palm Springs, California, October, 1999, pp. 10 - 16.
- Geist, R., Vernon, D., and Schalkoff, R., "Rendering Inversion in the Automated Construction of Virtual Environments," *Proc. 3rd ASCE Specialty Conf. on Robotics for Challenging Environments (ROBOTICS '98)*, Albuquerque, New Mexico, April, 1998, pp. 85 - 91.
- Geist, R., Schalkoff, R., Stinson, T., and Gurbuz, S., "Autonomous Virtualization of Real Environments for Telepresence Applications," *PRESENCE: Teleoperators and Virtual Environments*, 6:6(1997), MIT Press, pp. 645 - 657.
- N. Costescu, M. Loffler, E. Zergeroglu, and D. Dawson, "QRobot - A Multitasking PC Based Robot Control System", *Microcomputer Applications Journal Special Issue on Robotics*, Vol. 18, No. 1, 1999, pp. 13-22.
- W.E. Dixon, M.S. de Queiroz, D. M. Dawson, and F. Zhang, "Tracking Control of Robot Manipulators with Bounded Torque Inputs", *Robotica*, Vol. 17, pp. 121-129, 1999.
- N. Costescu, D. M. Dawson, and M. Loffler, "Qmotor 2.0 - A PC Based Real-Time Multitasking Graphical Control Environment", *IEEE Control Systems Magazine Applications*, Vol. 19, No. 3, pp. 68-76, Jun., 1999.
- E. Zergeroglu, W.E. Dixon, D. Haste, and D. M. Dawson, "A Composite Adaptive Output Feedback Tracking Controller for Robotic Manipulators", *Robotica*, Vol 17, pp. 591-600, 1999.
- E. Zergeroglu, W.E. Dixon, A. Behal, and D. M. Dawson, "Adaptive Set-Point Control of Robotic Manipulators with Amplitude-Limited Control Inputs", *Robotica*, Vol. 18, No. 2, pp. 171-181, 2000.
- M. Feemster, D. Dawson, A. Behal, and W. Dixon, "Tracking Control in the Presence of Nonlinear Dynamic Frictional Effects: Robot Extension," *Asian Journal of Control*, Vol. 1, No. 3, pp. 153-168, 1999.
- Y. Kim, F. Lewis, and D. Dawson, "Intelligent Optimal Control of Robotic Manipulators Using Neural Networks", *Automatica*, Vol. 36, pp. 1355-1364, 2000.

Conference Papers

- M. Feemster, P. Vedagarbha, D. M. Dawson, and D. Haste, "Adaptive Control Techniques for Friction Compensation", *Proc. of the American Control Conference*, Philadelphia, PA, Jun., 1998, pp. 1488-1492.
- N. Costescu and D. M. Dawson, "Qmotor 2.0 - A PC Based Real-Time Multitasking Graphical Control Environment", *Proc. of the American Control Conference*, Philadelphia, PA, Jun., 1998, pp. 1266-1270.
- W. Dixon, E. Zergeroglu, M. de Queiroz, and D. Dawson, "Global Output Feedback Tracking Control for Rigid-Link Flexible-Joint Robots", *Proc. of the International Conference on Robotics and Automation*, Leuven, Belgium, May, 1998, pp. 498-503.
- W.E. Dixon, M.S. de Queiroz, F. Zhang, and D.M. Dawson, "Tracking Control of Robot Manipulators with Bounded Torque Inputs", *Proc. of the 6th IASTED International Conference on Robotics and Manufacturing*, Banff, Canada, July, 1998, pp. 112-115.
- M.S. de Queiroz, D.M. Dawson, F. Zhang, and M. Agarwal, "Adaptive Control of Robot Manipulators with Controller/Update Law Modularity", *Proc. of the 6th IASTED International Conference on Robotics and Manufacturing*, Banff, Canada, July, 1998, pp. 116-119.
- W. Dixon, F. Zhang, D. Dawson, and A. Behal, "Global Robust Output Feedback Tracking Control of Robot Manipulators", *Proc. of the IEEE Conference on Control Applications*, Trieste, Italy, Sept., 1998, pp. 897-901.

- N. Costescu, M. Loffler, E. Zergeroglu, and D. Dawson, “QRobot - A Multitasking PC Based Robot Control System”, *Proc. of the IEEE Conference on Control Applications*, Trieste, Italy, Sept., 1998, pp. 892-896.
- E. Zergeroglu, W.E. Dixon, D. Haste, and D. M. Dawson, “A Composite Adaptive Output Feedback Tracking Controller for Robotic Manipulators”, *Proc. of the American Control Conference*, San Diego, CA, June, 1999, pp. 3013-3017.
- W.E. Dixon, E. Zergeroglu, D. M. Dawson, and M. W. Hannan, “Global Adaptive Partial State Feedback Tracking Control of Rigid-Link Flexible-Joint Robots”, *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, GA., Sept., 1999, pp. 281-286.
- E. Zergeroglu, D.M. Dawson, M. de Queiroz, and A. Behal, “Vision-based Nonlinear Tracking Controllers in the Presence of Parametric Uncertainty”, *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, GA., Sept., 1999, pp. 854 - 859.
- W.E. Dixon, E. Zergeroglu, D. M. Dawson, and M. W. Hannan, “Global Adaptive Partial State Feedback Tracking Control of Rigid-Link Flexible-Joint Robots”, *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, GA., Sept., 1999, pp. 281-286.
- W.E. Dixon, E. Zergeroglu, D. M. Dawson, and M. W. Hannan, “Experimental Video of a Controller for Flexible Joint Robots”, *Video Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, GA., Sept., 1999.
- E. Zergeroglu, M. de Queiroz, D.M. Dawson, and A. Behal, “Robust Visual Servo Control of Robot Manipulators”, *Proc. of the IEEE Conference on Decision and Control*, Phoenix, AZ, Dec., 1999, pp. 4137-4142.
- W.E. Dixon, I. D. Walker, D. M. Dawson, and J. P. Hartranft, “Fault Detection for Robotic Manipulators with Parameteric Uncertainty: A Prediction Error Based Approach”, *Proc. of the 1999 IEEE Conference on Robotics and Automation*, San Francisco, California, April, 2000, pp. 3628-3634.
- E. Zergeroglu, D.M. Dawson, I. D. Walker and A. Behal, “Nonlinear Tracking Control of Kinematically Redundant Robot Manipulators”, *Proc. of the American Control Conference*, Chicago, IL, June 2000, pp. 2513-2517.
- E. Zergeroglu, D. M. Dawson, Y. Fang, and A. Malatpure, “Adaptive Camera Calibration Control of Planar Robots: Elimination of Camera Space Velocity Measurements”, *Proc. of the IEEE Conference on Control Applications*, Anchorage, AK, September 2000, pp. 560-565.
- Z. Yao, N. P. Costescu, S. P. Nagarkatti, and D. M. Dawson, “Real- Time Linux Target: A MATLAB-Based Graphical Control Environment”, *Proc. of the IEEE Conference on Control Applications*, Anchorage, AK, September 2000, pp. 173-178.
- M. Loffler, N. Costescu, E. Zergeroglou, and D. Dawson, “Telerobotic Decontamination and Decommissioning with QRobot, a PC-Based Robot Control System”, *Proc. of the IEEE Conference on Control Applications*, Anchorage, AK, September 2000, pp. 24-29.
- E. Zergeroglu, D. M. Dawson, M.S. de Queiroz, and M. Krstic, “On Global Output Feedback Tracking Control of Robot Manipulators” *Proc. of the IEEE Conference on Decision and Control*, Sydney, Australia, Dec. 2000, accepted, to appear.

8 Interactions

- As indicated above, extensive conversations with INEEL personnel have occurred.
- In addition to presentations at the conferences (above), Robert Geist gave an invited presentation entitled, “Automated Construction of Virtual Environments,” to:
 - College of William & Mary
 - University of Alabama

- University of South Carolina
- Industrial Light & Magic (Lucas Digital Ltd.)

during the supported period of the grant.

Significant collaboration is underway with Dr. Barbara am Ende, National Institute of Standards and Technology. The surface reconstruction algorithms developed here for decommissioning appear to have great promise for the Wakulla Springs project in reconstructing underwater cave surfaces.

Clemson personnel worked with Barrett Technology to port our robotic software to the Barrett WAM (Whole Arm Manipulator). This software, known as the QMotor Robotic Toolkit (RTK), was demonstrated at the International Conference on Robotics and Automation (ICRA) in San Francisco in May 2000.

9 Future Work

Additional research is needed to evolve this process into working models and equipment that is ready for field deployment. The initial phase of this research has demonstrated the feasibility of such an approach; a follow-on effort would enhance the current technology and to transfer this technology from a laboratory prototype to field demonstration in actual DOE sites. Significant interest in this technology has been expressed by two DOE sites which have been identified as willing to cooperate in the transfer .

Specific topics include:

- Development and implementation of algorithms to allow real-time updating of world geometric and (gray-scale) photometric characterization ("re-virtualization") as D&D proceeds.
- Fusion of active surface information with passive -derived information to minimize the geometric interpolation induced by use of a finite number of light stripes (sampling).
- Development of robust algorithms for real-time, on the fly, accurate passive camera(s) recalibration or updating. To this end, we have arranged with Ascention Technologies for the loan of a prototype laserBIRD tracker which provides this pose information. We would work with Ascention Technologies on the refinement of this device; particularly important is the tradeoff between pose update rate and pose accuracy.
- Development of software to build polygonal surface representations directly from multiview point clouds, where the clouds are segmented only by target object. This allows volumes of objects to be estimated and used in D&D planning.
- Elimination of spurious geometric point solutions due to highly specular surfaces and reflected laser energy. This is common problem in industrial applications.
- Interface of the vision effort with monochromatic rendering approaches.
- Development of a portable, open-platform environment for experimentation.
- Implementation in real-time (via "fast" algorithms) and in a structure compatible with DOE specified software environments.

Thus, future research will include combining virtual reality research and vision-based site mapping techniques into an integrated, fielded system.